# МЕТОДИ ВИЯВЛЕННЯ ВРАЗЛИВОСТЕЙ У СМАРТ-КОНТРАКТАХ

Мокрій В.С., В'юхін Д.О.
Харківський національний університет радіоелектроніки, м. Харків, Україна

Смарт-контракти набули широкого застосування у DeFi (Decentralized Finance, децентралізовані фінанси), системах цифрових активів та управлінні ідентичністю. Публічність коду, незворотність транзакцій та неможливість змін після розгортання формують підвищені вимоги до безпеки [1]. Інцидент із The DAO (2016), де через вразливість reentrancy було виведено значний обсяг активів, унаочнює ці ризики.

Найпоширеніші вразливості смарт-контрактів - це reentrancy, integer overflow/underflow, порушення контролю доступу та front-running, що актуалізує потребу у комплексних підходах - статичному, динамічному та формальному аналізі [2].

**Метою доповіді** є комплексний аналіз методів виявлення вразливостей у смарт-контрактах та порівняльна оцінка інструментів їх реалізації за критеріями точності, повноти та практичної застосовності для підвищення рівня безпеки децентралізованих систем.

Статичний аналіз передбачає дослідження програмного коду без його виконання шляхом побудови абстрактних синтаксичних дерев, графів потоку керування та аналізу потоків даних. Такий підхід дозволяє ідентифікувати типові шаблони вразливостей ще на етапі розробки та легко інтегрується в CI/CD. До основних інструментів відносяться Slither (швидкий, точний аналіз для Solidity), Mythril (символьне виконання байт-коду EVM) та Oyente (перший публічний аналізатор EVM, орієнтований на виявлення front-running і reentrancy). Перевагою є швидкість та повнота охоплення коду, проте метод схильний до хибнопозитивних спрацювань [4].

Динамічний аналіз досліджує поведінку контракту під час виконання в тестовому середовищі та дозволяє виявляти логічні помилки, що проявляються лише за певних умов. Різновидом цього підходу є фаззінг - генерація псевдовипадкових вхідних даних з метою провокування некоректної поведінки; зокрема інструмент Echidna інтегрується з локальними тестовими мережами Ethereum для перевірки інваріантів контракту. Динамічний аналіз виявляє реально експлуатаційні вразливості, однак не гарантує повноти покриття [3].

Формальна верифікація передбачає математично доведену коректність коду відносно формальної специфікації за допомогою методів перевірки моделей та доведення теорем. Інструмент Certora Prover, що активно використовується у DeFi-протоколах (Aave, Compound), дозволяє отримати строгі гарантії коректності зміни стану контракту та відсутності несанкціонованого доступу. Незважаючи на найвищу надійність, метод є ресурсомістким і потребує спеціалізованих знань [4].

Порівняльна характеристика інструментів виявлення вразливостей представлена у таблиці 1.

Таблиця 1 - Характеристика інструментів виявлення вразливостей

| **Інстру-мент** | **Метод** | **Виявлювані вразливості** | **Точність** | **Орієнтовна вартість** |
|---|---|---|---|---|
| Slither | Статичний | Reentrancy, access control, tx.origin, shadowing | ~90% | Безкоштовно (open-source) |
| Mythril | Статичний / символьне виконання | Reentrancy, integer overflow, unsafe delegatecall | ~75% | Безкоштовно (open-source) |
| Oyente | Статичний / символьне виконання | Reentrancy, front-running, TOD | ~70% | Безкоштовно (open-source) |
| Echidna | Динамічний / фаззінг | Логічні помилки, порушення інваріантів | Висока | Безкоштовно; витрати на тестову мережу мінімальні |
| Certora Prover | Формальна верифікація | Некоректна зміна стану, несанкціонований доступ | Доведена | Від $0 до $5000+ |

Жоден із методів не є універсальним. Статичний аналіз швидкий, але дає хибнопозитивні результати. В свою чергу динамічний виявляє реальні вразливості без повного покриття. Формальна верифікація найнадійніша, але ресурсомістка.

Таким чином пропонується оптимальна стратегія - їх комбінування. Для більшості проектів найбільш практичним вибором є Slither: безкоштовний, ~90 % точності, легко інтегрується в CI/CD. Certora Prover при тарифі від $5 000 не виправдовує себе там, де безкоштовні інструменти закривають переважну більшість вразливостей.

**Список літератури**

1. Терещенко Г. Ю., Кириченко І. В. Аналіз і обґрунтування використання наявних блокчейн-рішень для захисту цифрових активів. ХНУРЕ, 2024.

2. Peschanenko V., Rud V. Overview of Vulnerabilities in Smart Contracts Written in Solidity. Journal of Information Technologies in Education (ITE), 2024.

3. Luu L., Chu D.-H., Olickel H., Saxena P., Hobor A. Making Smart Contracts Smarter. Proceedings ACM SIGSAC CCS, 2016.

4. Tsankov P., Dan A., Drachsler-Cohen D., Gervais A., Bünzli F., Vechev M. Securify: Practical Security Analysis of Smart Contracts. Proceedings ACM SIGSAC CCS, 2018.